(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 339 782 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 89302579.1

(51) Int. Cl.4: G06F 13/36

(22) Date of filing: 16.03.89

(30) Priority: 28.04.88 JP 104471/88

(43) Date of publication of application: 02.11.89 Bulletin 89/44

(84) Designated Contracting States: DE FR GB

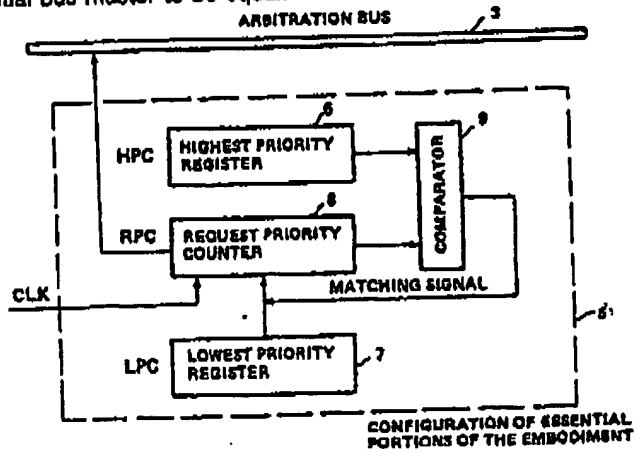
(71) Applicant: International Business Machines Corporation
Old Orchard Road
Armonk, N.Y. 10504(US)

(72) Inventor: Ohba, Nobuyuki
MatsuoCorpo 202,1-170 Sakato Takatsu-ku
Kawasaki-shi Kanagawa-ken(JP)

(74) Representative: Grant, Iain Murray
IBM United Kingdom Limited Intellectual Property Department Hursley Park
Winchester Hampshire SO21 2JN(GB)

(54) Shared bus data processing systems.

(57) A shared bus data processing system in which a plurality of potential bus masters can contend for time slot ownership on the shared bus on a priority basis is disclosed, wherein the current priority of each master is determined by comparing a count, adjusted monotonically in synchronism with the available time slots, is compared with one of a pair of stored boundary values for that master and, if that master is contending, is the priority value for that master if the current count is found to lie within the comparison boundary value, else is deemed to be the other stored boundary value. The stored boundary values and the initial count for each potential bus master are all programmable whereby "round robin" priority is obtainable by setting the stored boundary difference equal to the number of potential bus masters in the round robin and the initial counts to lie within the boundary difference and to be representative of the position within the round robin priority order of the particular potential bus master to which relates and fixed priority is obtainable by setting the two stored boundary values for a potential bus master to be equal.



CONFIGURATION OF ESSENTIAL PORTIONS OF THE EMBODIMENT

FIG. 1

EP 0 339 782 A2

SHARED BUS DATA PROCESSING SYSTEMS

The present invention relates to shared bus data processing systems in which a plurality of potential bus masters are connected to the shared bus and provision is made to provide bus use priority to each bus master in a flexible and optimum manner, enabling the enhancement of bus use efficiency.

In a system in which a plurality of bus masters are connected by a shared bus, such as a multiprocessor system or a channel subsystem, a bus use request may be issued from more than one bus master at the same time. In that case, the bus use must be permitted by selecting one bus master according to the priority of each bus master. With respect to the priority assignment, the following two methods have so far been used.

(1) Fixed priority method

Previously fixed priorities are set in each of the bus masters, and if a bus use request is issued simultaneously from more than one bus master, the bus use is granted to the bus master that has the highest priority. This method is the easiest one to provide, but it has a shortcoming in that the use efficiency of the bus will be decreased when there is a large number of bus masters. Also, a situation may occur in which bus masters of lower priorities cannot get a bus at all.

(2) Round robin method

This is a method in which each of the bus masters is given a serial number and priority is determined according to its values. This serial number is changed according to the use of the bus to keep the bus use uniform. However, the bus master includes a type in which the data transfer speed is fixed as in a DMA (Direct Memory Access) controller and the waiting time after issuing a bus use request is limited (for example, if reading-out of data from a disk is carried out by the DMA transfer, the bus use permission must be received and the transfer must be completed within a prescribed time after sending out the bus request), and a type in which the transfer speed and period will change as in a CPU (Central Processing Unit). Even if a single bus master of the type for which the time till the transfer completion is limited, such as a DMA controller is included, this method cannot be used.

As an example of a bus system using the above fixed priority method, MULTIBUS II (a trademark) of the Intel Corporation is cited. Further, Motorola's MC88452 can select and use either the fixed priority method or the round robin method. However, it cannot use both methods simultaneously.

Incidentally, the following patent references are pertinent.

Published Unexamined Patent Application (PUPA) No. 62-9454 official gazette: In a system in which the bus arbitration function is distributed among the bus masters, the fixed priority method is implemented by providing each bus master with a register for holding a priority.

PUPA No. 62-187955 official gazette: In a similar system, the round robin method is implemented by providing a counter.

However, the prior art discloses nothing which appears to provide the potential for selectively obtaining the advantages of both the round robin method and the fixed priority method.

Accordingly, the present invention provides a shared bus data processing system in which a plurality of potential bus masters can contend for time slot ownership on the shared bus on a priority basis, wherein the current priority of each master is determined by comparing a count, adjusted monotonically in synchronism with the available time slots, is compared with one of a pair of stored boundary values for that master and, if that master is contending, is the priority value for that master if the current count is found to lie within the comparison boundary value, else is deemed to be the other stored boundary value.

In one form of data processing system according to the present invention and disclosed hereinafter, each bus master holds a priority code representing the bus use priority which currently assigned to it, and it outputs that code to a bus arbitration circuit when requesting the bus. The bus arbitration circuit may be one that collectively performs the arbitration as a dedicated unit, or may be one that is distributed and held among the respective bus masters. In the present invention, each bus master has a highest priority register indicating the highest priority which it can possess, a lowest priority register indicating the lowest priority, and a request priority counter indicating the priority which it is currently provided with. And the counter is either incremented or decremented for each bus arbitration sequence. In a construction in which the request

priority in the counter approaches the highest priority by incrementing or decrementing, the request priority is compared with the lowest priority and, when the two are equal to the lowest priority in the lowest priority register is transferred to the counter. Conversely, in a construction in which the request priority approaches the lowest priority, the request priority is compared with the lowest priority and, when the two are equal, the highest priority in the highest priority register is transferred to the counter.

In such an arrangement, the highest priority, the lowest priority and the contents of each bus master can arbitrarily be established, and the bus arbitration can be conducted in various manners according to how they are chosen. For example, the fixed priority method can be implemented by, setting both the highest priority and the lowest priority of each bus master to be fixed priority provided to that bus master. Also, the round robin method can be implemented by setting the same highest priority and the same lowest priority in each bus master, and if the difference between the highest priority is made larger than the lowest priority by a number equal to the number of bus masters, and if each of the initial values in the counters of the respective bus masters is different. It is also possible that the fixed priority method may be used for part of the bus masters with the remaining bus masters using the round robin method.

The present invention will be described further by way of example with reference to an embodiment thereof as illustrated in the accompanying drawings, in which:

Fig. 1 is a block diagram showing the essential portions of an embodiment of the present invention;

Fig. 2 is a block diagram showing the overall configuration of the above-mentioned embodiment;

Figs. 3 to 6 are illustrations for explaining the operation of the above-mentioned embodiment;

Fig. 7 is a block diagram showing in more detail the essential portions of the above-mentioned embodiment; and

Fig. 8 is a time chart for explaining Fig. 7.

Fig. 2 shows as a whole an embodiment of the present invention, and in this figure, n bus masters BM1 to BMn - potential, since only one at a time can be an actual bus master, but the term is dropped hereinafter in the description for simplicity - are connected to a shared bus system. The shared bus system has an external data bus 1, an external address bus 2 and an arbitration bus 3. Each bus master BM1 to BMn, holds a priority code representing the bus use right it is currently provided with, and outputs that code to the arbitration bus 3 when using the bus system. Each bus master, BM1 to BMn is provided with an arbitration control circuit (designated by 4 in Fig. 7) for distributive control of the bus system. That is, when each bus master BM1 to BMn takes part in the bus arbitration by outputting the priority code, it uses the bus system when the request priority larger than its own priority code.

Fig. 1 shows the request priority determination circuit 5 of the bus master BM1. The other bus masters, BM2 to BMn also have a similar request priority determination circuit 5. In Fig. 1, the priority determination circuit 5 comprises a highest priority register 6, a lowest priority register 7, a request priority counter 8 and a comparator 9. In the highest priority register 6, a priority code HPC representing the highest priority that the bus master BM1 can take is stored, and in the lowest priority register 7, a priority code LPC representing the lowest priority that the bus master BM1 can take is stored. Incidentally, in the bus arbitration of this embodiment, the priority becomes higher as the priority code decreases.

Also, a clock CLK is inputted to the request priority counter 8, which counts down in response to this clock CLK. One clock CLK is generated when a bus master acquires the use of the bus and starts to use the bus. The contents of the request priority counter 8 are supplied as a priority code RPC of the request priority to the arbitration bus 3 and the comparator 9. The comparator 9 compares the priority code RPC of the request priority and the priority code HPC of the highest priority, and outputs a matching signal when they are equal. This matching signal is supplied to the counter 8, and based on this matching signal, the priority code LPC of the lowest priority of the lowest priority register 7 is transferred to the request priority counter 8 at the timing of the next clock CLK.

In such a configuration, the priority code RPC of the request priority is decremented by one each time the arbitration is completed, and returns to the priority code LPC of the lowest priority when it reaches the priority code HPC of the highest priority. This means that the request priority circulates between the highest priority and the lowest priority.

Now, the operation of this embodiment is described by giving examples.

For simplicity, a system is considered wherein four bus masters BM1 to BM4 are connected to one bus. It is now desired to always give the highest priority (fixed priority code 0) to the bus master BM1 and to give priority (variable priority code 1, 2 and 3) to the remaining bus masters BM2, BM3 and BM4. In that case, the highest priority code HPC, the lowest priority code LPC and the request priority code RPC are established as shown in Fig. 3. That is, by setting "0" in the HPC, LPC and RPC, the bus master BM1 is made ready to issue a bus use request of a priority code 0 at any time. For the bus masters BM2, BM3 and

BM4, the HPC is set to "1" and the LPC is set to "3", and the initial values are set so that the bus request priority code is either "1", "2" or "3" and a duplication of the RPC is avoided.

First, it is supposed that a bus use request has concurrently been issued from the bus masters BM2 and BM3. The bus master BM2 outputs its own current priority code 1 (the value of RPC) on the bus. On the other hand, the bus master BM3 also outputs its own current priority code 2. Since it has been arranged that the priority becomes higher as the priority code decreases, the bus master BM2 will acquire and use the bus. After that, the values of RPC in the bus masters BM1 to BM4 are as shown in Fig. 4.

Then, it is supposed that a bus request has been issued again from the bus masters BM2 and BM3. Since, this time, the RPC of the bus master BM2 is "3" and the RPC of the bus master BM3 is "1", the bus master BM3 will acquire the bus. And the RPC of each bus master BM1 to BM4 changes after the arbitration and is set as shown in Fig. 5.

If, in the above example, the bus master BM1 also requested the bus, the bus master BM1 could always acquire the bus with the highest priority (priority code 0) because its RPC, HPC and LPC have been all set to "0". The changes with time of the RPC values in the bus masters BM1 to BM4 are shown in Table 1.

Table 1

Changes of Priority Codes with Time				
Arbitration	Value of Priority Code (RPC)			
	Bus Master BM1	Bus Master BM2	Bus Master BM3	Bus Master BM4
1st	0	1	2	3
2nd	0	3	1	2
3rd	0	2	3	1
4th	0	1	2	3
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

As another example, a case is considered in which the bus masters BM1 to BM4 are provided with fixed priorities (bus master BM1 > bus master BM2 > bus master BM3 > bus master BM4; however, the code magnitude is in the reverse order to this). In this case, as shown in Fig. 6, the HPC, LPC and RPC of the bus master BM1 are set to "0", those of the bus master BM2 to "1", those of the bus master BM3 to "2", and those of the bus masters BM4 to "3". With this, the respective bus masters BM1 to BM4 always output priority codes of "0", "1", "2", and "3".

In addition, in this embodiment, each bus master BM1 to BM4 may be used to rewrite the HPC, LPC and RPC during the system operation. Incidentally, in this example, the respective bus masters were made to have a distributed arbitration control function, but a configuration may also be used in which the arbitration control is collectively performed.

Fig. 7 shows the details of the request priority determination circuit 5 of the example in Fig. 1, along with an arbitration control circuit 4 and a CPU 10. In this figure, the portions corresponding to Fig. 1 are assigned the corresponding symbols, and the description thereof is omitted.

In Fig. 7, the respective signals are as follows:

HPC_LOAD: A signal for loading data on the CPU data bus to the highest priority register.

RPC_LOAD: A signal for loading data on the CPU data bus to the request priority counter.

LPC_LOAD: A signal for loading data on the CPU data bus to the lowest priority register.

Count_down_clock: A clock for changing the value of the request priority counter for each bus use. Similar to the CLK in Fig. 1.

CPU_RW: A signal for indicating the read-out or write-in of the CPU.

In this example, the desired data is transferred respectively to the highest priority register 6, the lowest priority register 7 and the request priority counter 8, based on the signals HPC__LOAD, LPC__LOAD and RPC__LOAD. After that, the RPC decreases for each arbitration sequence and, when the RPC equals the HPC, the RPC returns to the LPC. This is shown in Fig. 7. As apparent from this figure, in the case that a matching signal is only outputted from the comparator 9, data is not transferred to the request priority counter 8. If a matching signal is generated and the clock Count__down__clock is inputted to the counter 8, then data is transferred for the first time. By noting this point, it can be seen that the contents of the counter 8, or the request priority will change as follows. That is, if RPC=HPC at the clock CLK1, RPC=HPC is maintained also at the next arbitration time, and RPC=HPC is outputted to the arbitration bus 3 if taking part in the arbitration. At the clock CLK2, right after the completion of this arbitration, the LPC is transferred to the request priority counter 8. After this, the RPC is approaching the HPC in response to the subsequent arbitrations.

To summarise, this arrangement provides one form of shared bus data processing system in which a plurality of potential bus masters can contend for time slot ownership on the shared bus on a priority basis is disclosed, wherein the current priority of each master is determined by comparing a count, adjusted monotonically in synchronism with the available time slots, is compared with one of a pair of stored boundary values for that master and, if that master is contending, is the priority value for that master if the current count is found to lie within the comparison boundary value, else is deemed to be the other stored boundary value. The stored boundary values and the initial count for each potential bus master are all programmable whereby "round robin" priority is obtainable by setting the stored boundary difference equal to the number of potential bus masters in the round robin and the initial counts to lie within the boundary difference and to be representative of the position within the round robin priority order of the particular potential bus master to which relates and fixed priority is obtainable by setting the two stored boundary values for a potential bus master to be equal. Each potential bus master incorporates a first and a second boundary value register, a settable counter synchronised with the bus time slots, a comparator for comparing the contents of the first register with the contents of the counter, means for transferring the contents of the second register to the counter IFF the contents of the first register and the counter compare and means for transmitting the contents of the counter as the priority value for that potential bus master when contending. The bus time slot length is equal to one bus clock beat and the count adjustment period is synchronised to the bus clock.

It will be obvious to one skilled in the art that both the bus ownership arbitration and the current priority

value generation can be centralised, as for example, into one, only, of the potential bus masters.

Claims

1. A shared bus data processing system in which a plurality of potential bus masters can contend for time slot ownership on the shared bus on a priority basis, wherein the current priority of each master is determined by comparing a count, adjusted monotonically in synchronism with the available time slots, is compared with one of a pair of stored boundary values for that master and, if that master is contending, is the priority value for that master if the current count is found to lie within the comparison boundary value, else is deemed to be the other stored boundary value.

2. A system as claimed in claim 1 wherein the stored boundary values and the initial count for each potential bus master are all programmable whereby "round robin" priority is obtainable by setting the stored boundary difference equal to the number of potential bus masters in the round robin and the initial counts to lie within the boundary difference and to be representative of the position within the round robin priority order of the particular potential bus master to which relates and fixed priority is obtainable by setting the two stored boundary values for a potential bus master to be equal.

3. A system as claimed in either preceding claim wherein the priority determination is centralised in one of the potential bus masters.

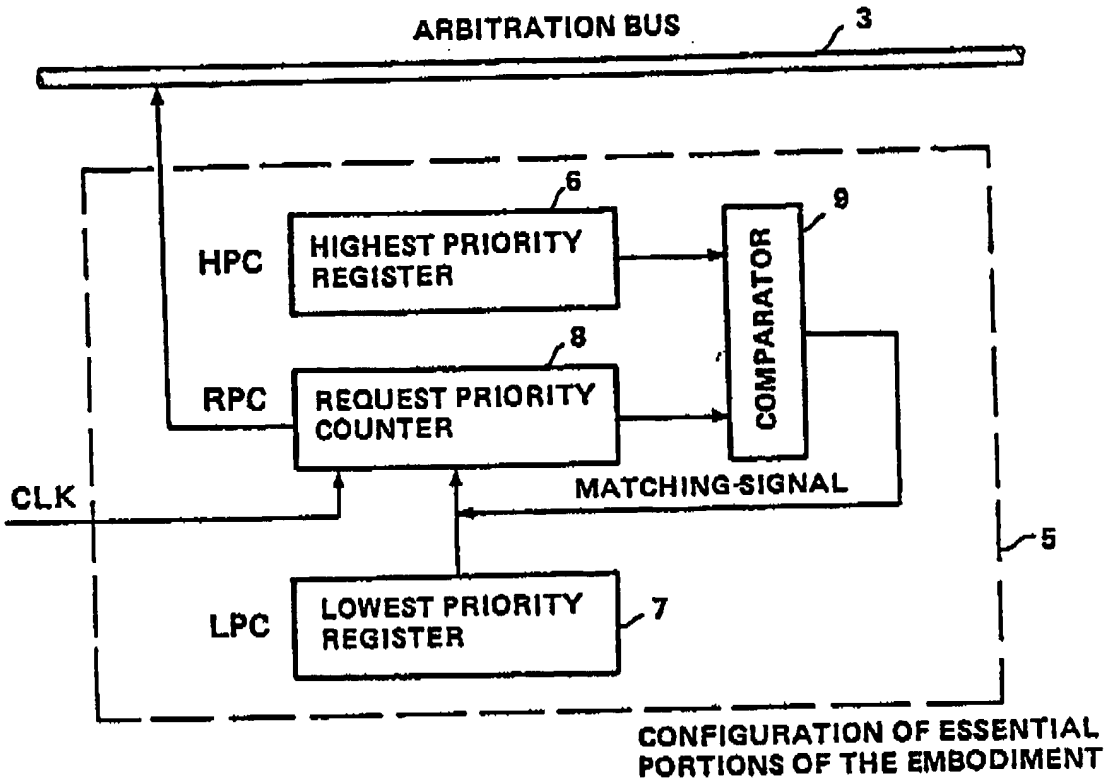
4. A system as claimed in claim 2 wherein each potential bus master incorporates a first and a second boundary value register, a settable countersynchronised with the bus time slots, a comparator for comparing the contents of the first register with the contents of the counter, means for transferring the contents of the second register to the counter IFF the contents of the first register and the counter compare and means for transmitting the contents of the counter as the priority value for that potential bus master when contending.

5. A system as claimed in Claim 4 wherein a bus arbitration circuit for collectively receiving the priority values outputted from the contending bus masters to collectively arbitrate current ownership of the shared bus is provided independently of the bus masters.

6. A system as claimed in Claim 4 wherein each of the potential bus masters has a bus arbitration facility receiving priority values from all other contending potential bus masters for determining whether it can gain current ownership of the shared bus on the basis of the pre-eminence of its own priority value relative to the received priority values.

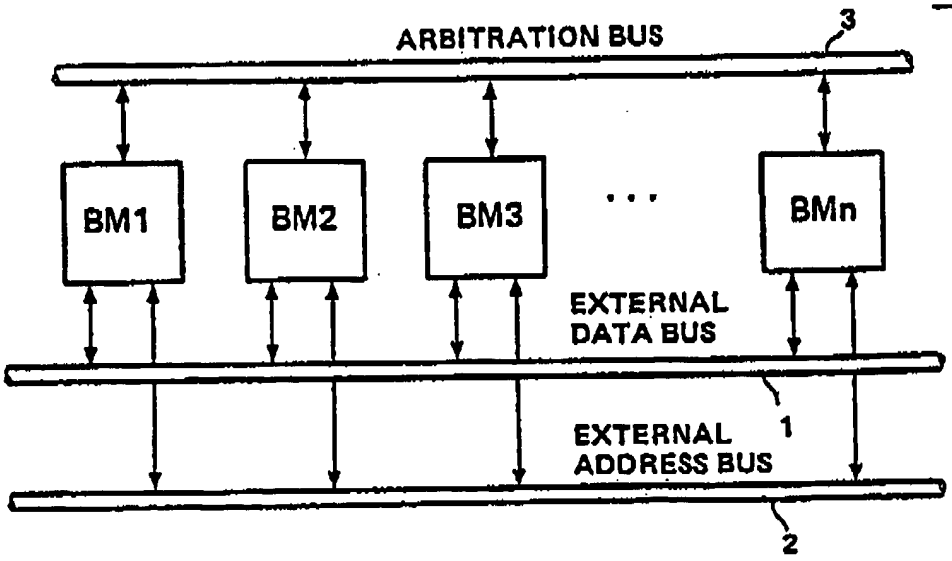
7. A system as claimed in any preceding Claim wherein the bus time slot length is equal to one bus clock beat and the count adjustment period is synchronised to the bus clock.

8. A priority determination circuit used in a system as claimed in any preceding claim in or for each potential bus master, the circuit comprising:
a first register for storing a first priority code;
a second register for storing a second priority code;
a counter for receiving a predetermined number of count pulses;
a comparing means for comparing the content of the counter with the first priority code in the first register;
a means for transferring the second priority code in the second register to the counter on the basis of the comparison result of the comparing means; and
means for outputting the contents of the counter as a coded representation of the priority value for that potential bus master, if contending.



CONFIGURATION OF ESSENTIAL PORTIONS OF THE EMBODIMENT

FIG. 1



SYSTEM OF EMBODIMENT

FIG. 2

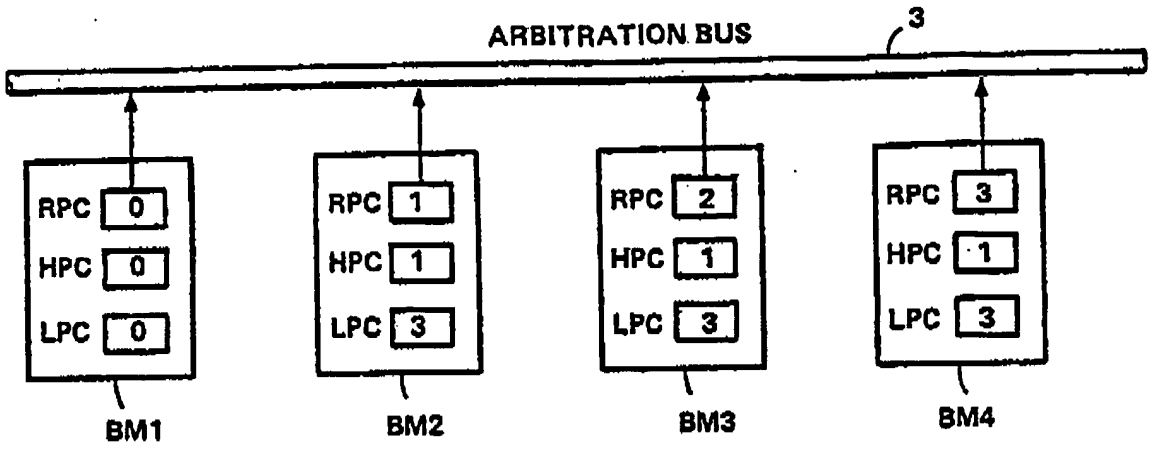

ARBITRATION BUS
3
RPC 0
HPC 0
LPC 0
BM1
RPC 1
HPC 1
LPC 3
BM2
RPC 2
HPC 1
LPC 3
BM3
RPC 3
HPC 1
LPC 3
BM4
INITIAL SETTING STATE

FIG. 3

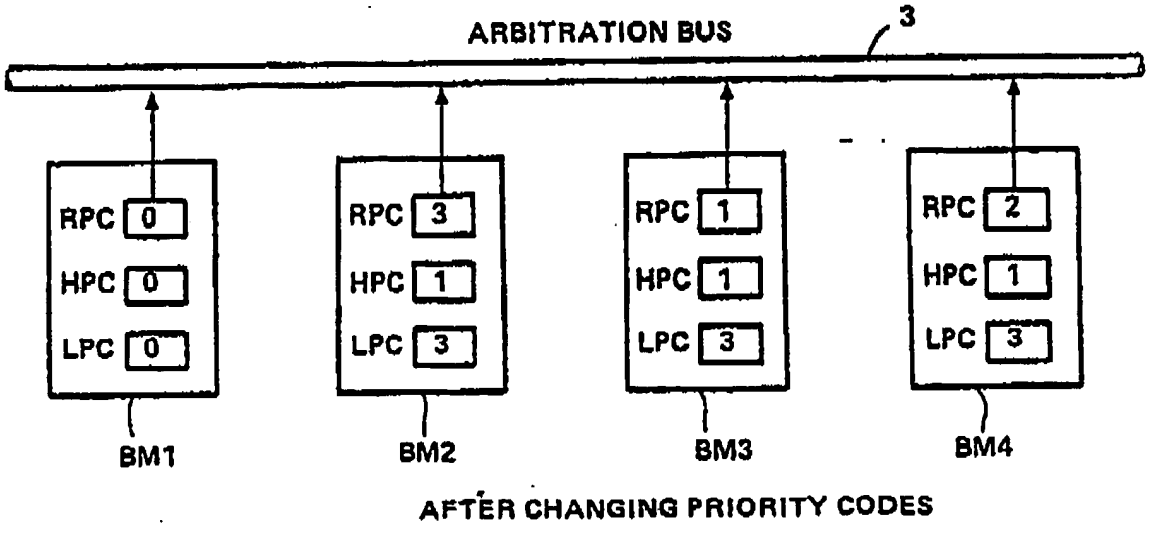
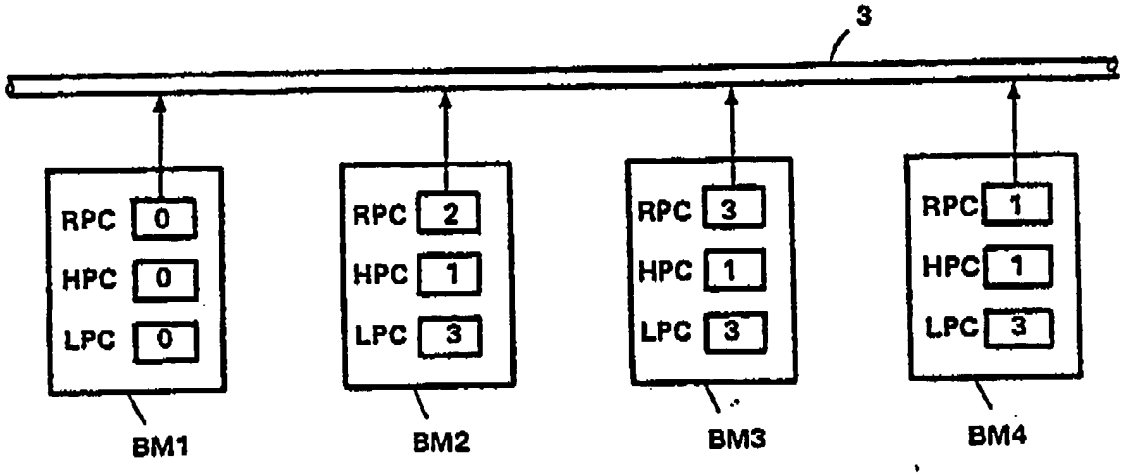
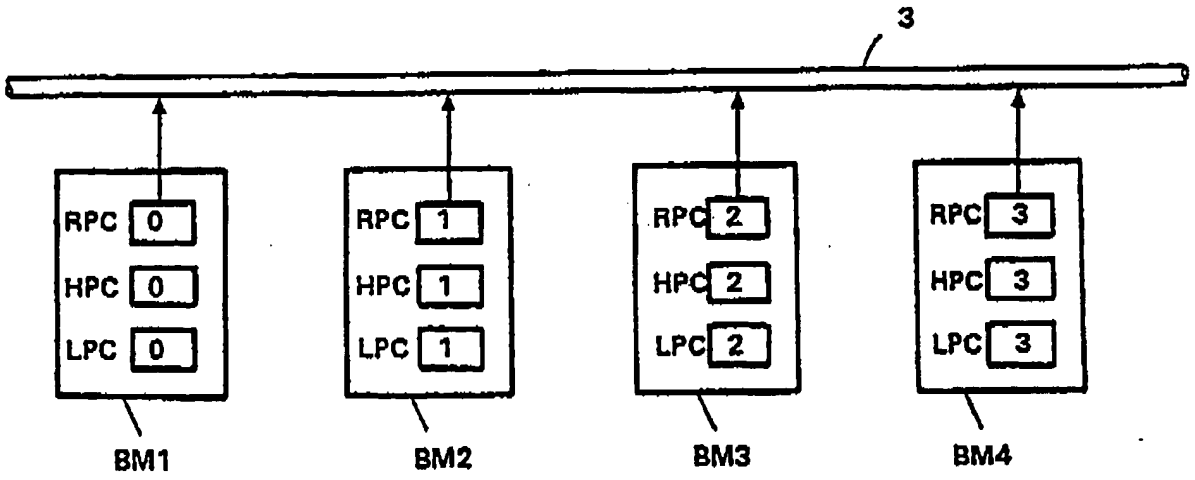

ARBITRATION BUS
3
RPC 0
HPC 0
LPC 0
BM1
RPC 3
HPC 1
LPC 3
BM2
RPC 1
HPC 1
LPC 3
BM3
RPC 2
HPC 1
LPC 3
BM4
AFTER CHANGING PRIORITY CODES

FIG. 4



AFTER RE-CHANGING

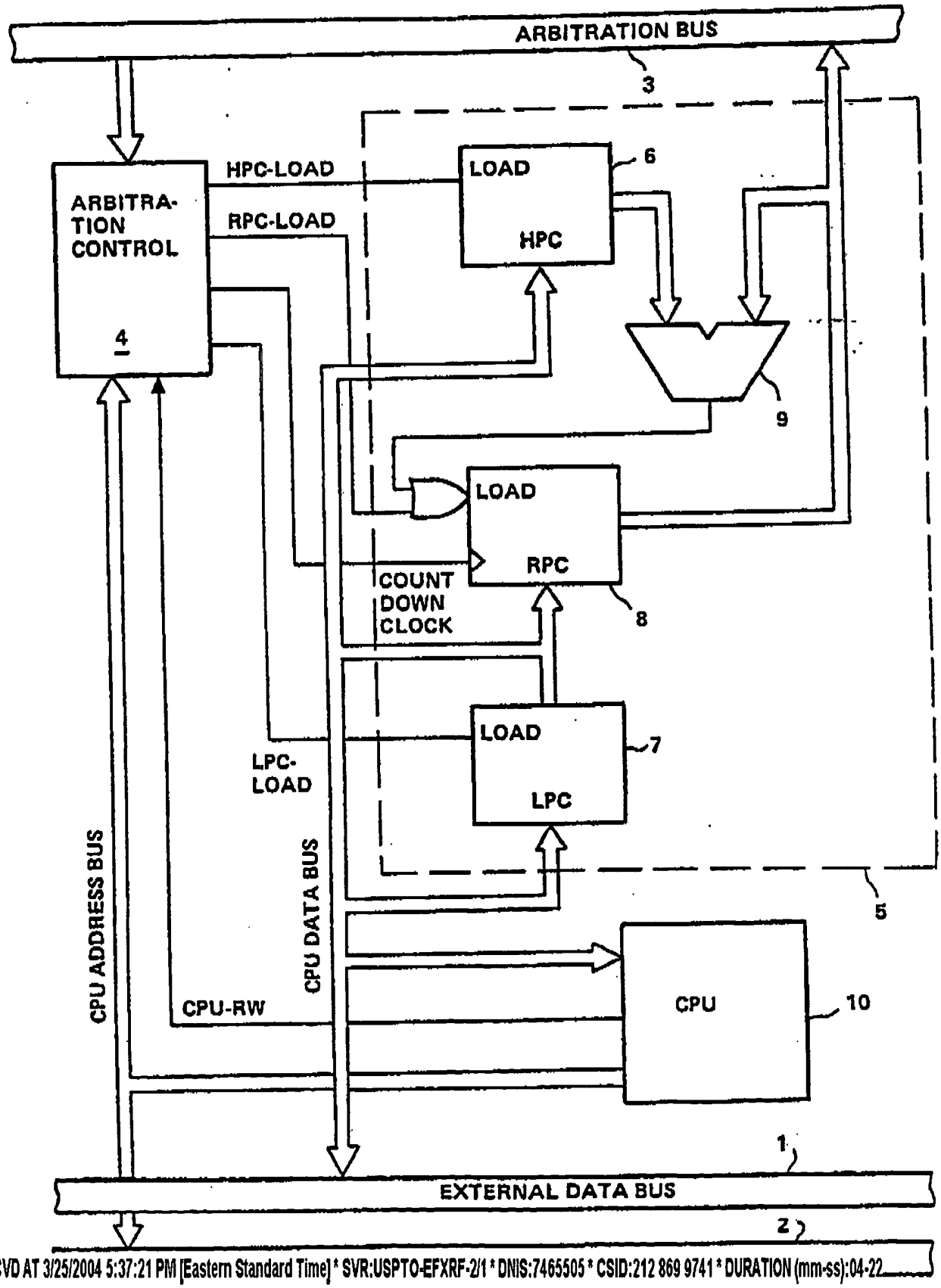
FIG. 5



ANOTHER INITIAL SETTING STATE

FIG. 6

FIG. 7


ARBITRATION BUS
3
ARBITRA-
TION
CONTROL
4
HPC-LOAD
RPC-LOAD
LOAD
HPC
6
9
LOAD
RPC
8
COUNT
DOWN
CLOCK
LOAD
LPC
7
LPC-
LOAD
CPU ADDRESS BUS
CPU DATA BUS
CPU-RW
5
CPU
10
1
EXTERNAL DATA BUS
2

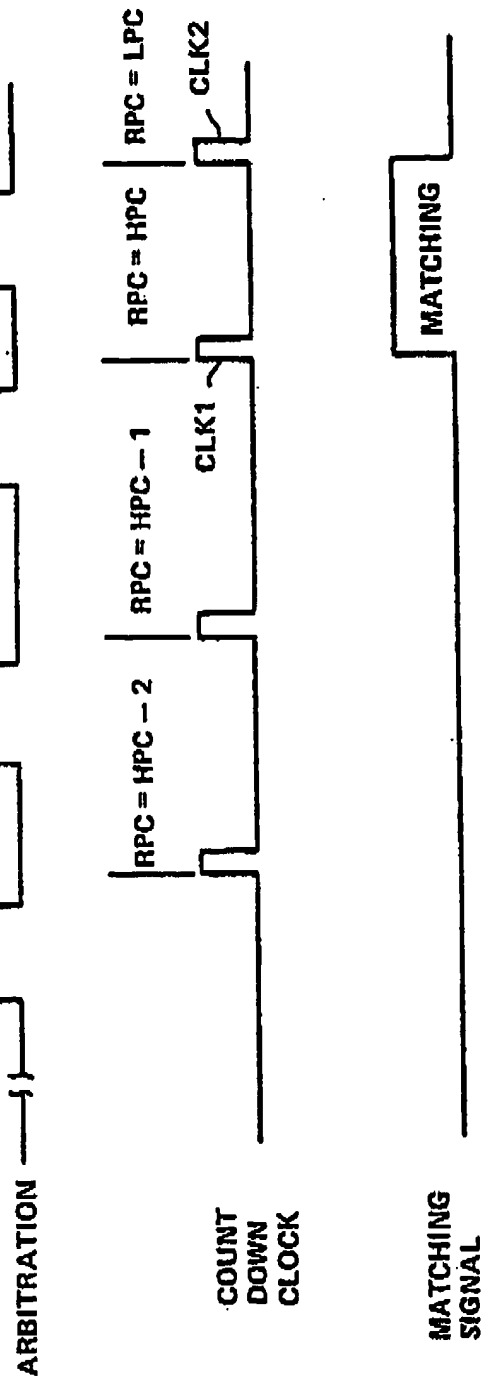

ARBITRATION
ARBITRATION
ARBITRATION
ARBITRATION
ARBITRATION
RPC = HPC – 2
RPC = HPC – 1
RPC = HPC
RPC = LPC
CLK1
CLK2
COUNT DOWN CLOCK
MATCHING
MATCHING SIGNAL

FIG. 8